

WO9710604

Publication Title:

Method of and architecture for controlling system data with automatic wear leveling in a semiconductor non-volatile mass storage memory

Abstract:

A semiconductor non-volatile mass storage memory is partitioned into user files and system files. The system files partition is further subdivided into clusters, each cluster having a plurality of sectors. Each cluster stores the system file for a single predetermined LBA. As the information within the LBA is changed, the new information is written into an empty sector within the cluster. Once the cluster is filled, the system either erases for recycling the cluster or preferably locates an empty cluster and repeats the process with that new cluster. Once all the clusters are filled, all clusters containing old data are erased for recycling.

Data supplied from the esp@cenet database - <http://ep.espacenet.com>

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G11C 16/06	A1	(11) International Publication Number: WO 97/10604
		(43) International Publication Date: 20 March 1997 (20.03.97)

(21) International Application Number: PCT/US96/14750

(22) International Filing Date: 12 September 1996 (12.09.96)

(30) Priority Data:
08/527,484 13 September 1995 (13.09.95) US

(71) Applicant: LEXAR MICROSYSTEMS INC. [US/US]; 47421 Bayside Parkway, Fremont, CA 94538 (US).

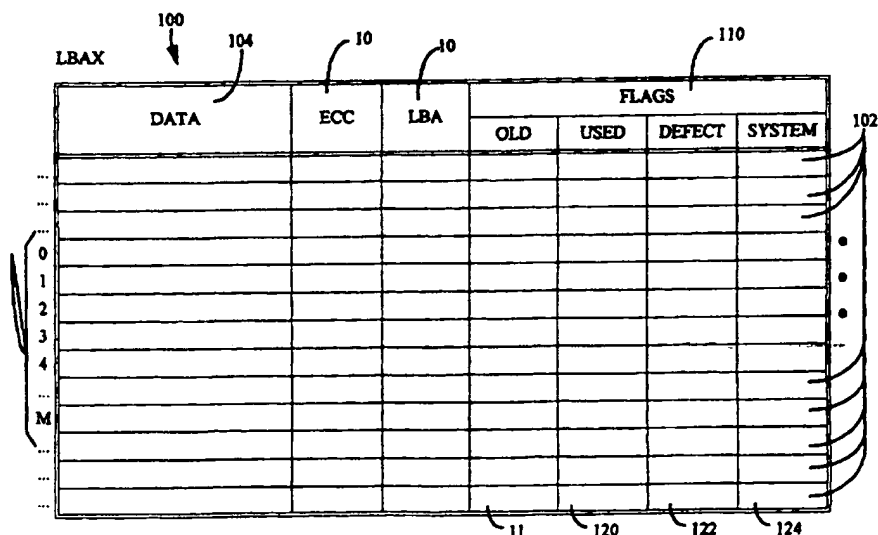
(72) Inventors: ESTAKHRI, Petro; 7966 Foothill Knolls, Pleasanton, CA 94566 (US). ASSAR, Mahmud; 14525 Shadowlane Court, Morgan Hill, CA 95037 (US). REID, Robert, A.; 44224 Ibero Way, Fremont, CA 94539 (US). IMAN, Berhanu; 946 Iris Avenue, Sunnyvale, CA 94086 (US).

(74) Agent: IMAM, Maryam; Lexar Microsystems, 47421 Bayside Parkway, Fremont, CA 94538 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published*With international search report.*

(54) Title: METHOD OF AND ARCHITECTURE FOR CONTROLLING SYSTEM DATA WITH AUTOMATIC WEAR LEVELING IN A SEMICONDUCTOR NON-VOLATILE MASS STORAGE MEMORY

**(57) Abstract**

A semiconductor non-volatile mass storage memory is partitioned into user files and system files. The system files partition is further subdivided into clusters, each cluster having a plurality of sectors. Each cluster stores the system file for a single predetermined LBA. As the information within the LBA is changed, the new information is written into an empty sector within the cluster. Once the cluster is filled, the system either erases for recycling the cluster or preferably locates an empty cluster and repeats the process with that new cluster. Once all the clusters are filled, all clusters containing old data are erased for recycling.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

**METHOD OF AND ARCHITECTURE FOR CONTROLLING
SYSTEM DATA WITH AUTOMATIC WEAR LEVELING IN A
SEMICONDUCTOR NON-VOLATILE MASS STORAGE MEMORY**

Field of the Invention

This invention relates to the field of semiconductor mass storage architectures and methods of operating therefor. More particularly, this invention relates to the field of controlling system data with automatic wear leveling in a semiconductor non-volatile mass storage system.

5

Background of the Invention

Computers conventionally use rotating magnetic media for mass storage of documents, data, programs and information. Though widely used and commonly accepted, such hard disk drives suffer from a variety of deficiencies. Because of the rotation of the disk, there is an inherent latency in extracting information from a hard disk drive.

10

Other problems are especially dramatic in portable computers. In particular, hard disks are unable to withstand many of the kinds of physical shock that a portable computer will likely sustain. Further, the motor for rotating the disk consumes significant amounts of power decreasing the battery life for portable computers.

15

Solid state memory is an ideal choice for replacing a hard disk drive for mass storage because it can resolve the problems cited above. Potential solutions have been proposed for replacing a hard disk drive with a semiconductor memory. For such a system to be truly useful, the memory must be non-volatile and alterable. The inventors have determined that FLASH memory is preferred for such a replacement.

20

FLASH memory is a single transistor memory cell which is programmable through hot electron injection or source injection and erasable through Fowler-Nordheim tunneling. The programming and erasing of such a memory cell requires current to pass through the dielectric surrounding a floating gate electrode. Because of this, such types of memory have a finite number of erase-write cycles. Eventually, the dielectric will fail. Typically, manufacturers of FLASH cell devices specify the limit for the number of erase-write cycles as between 100,000 and 1,000,000.

25

Another difficulty with some FLASH technologies is the existence of over-erase

problems. Devices manufactured using such technologies need to be programmed before erasing to improve uniformity amongst all cells. They also have to be verified for successful erase.

One requirement for a semiconductor mass storage device to be successful is that its
5 use in lieu of a rotating media hard disk mass storage device be transparent to the computer's operating system and the user of a system using such a device. In other words, the designer or user of a computer incorporating such a semiconductor mass storage device could simply remove the hard disk and replace it with a semiconductor mass storage device. All presently available commercial software should operate on a system employing
10 such a semiconductor mass storage device without the necessity of any modification.

Several architectures for a solid state mass storage device have been suggested in the prior art. In particular, two of the present inventors were co-inventors in solutions that are disclosed in U.S. Patent Application serial number 08/038,668 filed on March 26, 1993, now U.S. Patent 5,388,083, issued February 7, 1995, U.S. Patent Application serial number
15 08/037,893 also filed on March 26, 1993, U.S. Patent Application serial number 131,495 filed October 4, 1993, and U.S. Patent Application serial number (not yet assigned, Entitled: DIRECT LOGICAL BLOCK ADDRESSING FLASH MEMORY MASS STORAGE ARCHITECTURE, inventors: Estakhri and Assar) filed July 31, 1995. These four patent documents are incorporated herein by reference. These four patent documents
20 are commonly owned and commonly assigned to the same entity as the present application.

Conventional computer systems include a mass storage memory facility for storing user information such as data and applications files (user files). In addition to storing such data and applications files, conventional systems also store non-user information in the mass storage memory facility for use by the computer system in controlling the general
25 operation of the computer and its related peripheral systems (system files). In DOS based systems, such system information includes the boot sector, the FAT tables and the directory information.

According to such conventional systems, the system files are routinely changed significantly more often than the user files. Indeed, for each write of user files, the FAT
30 tables and directory are accessed and written. Within a FLASH memory device, the write and erase operation take the most time to perform. Accordingly, it would be advantageous to not have to perform an erase operation every time a system file is updated. Also, most

FLASH memory devices will erase multiple blocks simultaneously. Because, most system file updates are performed in a single sector format, it is desirable to handle system files differently than the data files within the FLASH memory in order to take advantage of a multi-sector erase function. Other approaches have been proposed for handling the writing of information onto a semiconductor mass storage system. However, none of these approaches teaches a method or apparatus which specifically addresses the unique requirements of system files.

Summary of the Invention

10 A semiconductor non-volatile mass storage memory is partitioned into user files and system files. The system files partition is further subdivided into clusters, each cluster having a plurality of sectors. Each cluster stores a sector of the system file for a single predetermined LBA. As the information within the LBA is changed, the new information is written into an empty sector within the cluster. Once the cluster is filled, the system either
15 erases for recycling the cluster or preferably locates another empty cluster and repeats the process with that new cluster. Once a cluster is filled, it is either erased or flagged as "old" for erasing later.

Brief Description of the Drawings

20 Figure 1 shows a schematic representation of a semiconductor non-volatile mass storage memory according to the present invention.

Figure 2 shows a schematic representation of a partitioning of the semiconductor non-volatile mass storage memory according to the present invention.

Figure 3 shows a schematic representation of the cluster arrangement within the
25 system file partition according to one embodiment of the present invention.

Figure 4 shows a schematic representation of the cluster arrangement within the system file partition according to the preferred embodiment of the present invention. Figure 5 and Figure 6 show the arrangement of Figure 4 after a sequence of operations according to the invention.

30 Figure 7 shows a schematic representation of the block RAM for the present invention.

Figure 8 illustrates an example of how the entries within the block RAM table are changed in response to a sequence of operations.

Figure 9 illustrates a block diagram schematic of a mass storage memory control circuit of the present invention.

5

Detailed Description of the Preferred Embodiment

The present invention is for a method of and an architecture for controlling system data with automatic wear leveling in a semiconductor non-volatile mass storage memory with improved performance. It will be well understood that the mass storage memory will be coupled to a computer or other digital system which can access the information stored in the mass storage memory. Figure 1 shows a schematic representation of a cluster within a semiconductor non-volatile mass storage memory according to the present invention. The memory structure is preferably formed of FLASH memory cells. However, it will be apparent to one of ordinary skill in the art that other memory types can be used according to the teachings of the present invention.

15

The mass storage memory 100 is divided into a plurality of addressable memory portions 102. In the preferred embodiment, each such memory portion 102 is a sector such as addressable in a DOS type application. Other convenient memory sizes can be used. For convenience of understanding, the term sector 102 will be used to discuss these memory portions. Within the mass storage memory 100, the sectors 102 are grouped into clusters 116. Each cluster 116 includes M sectors 102, where M is any convenient size, but is preferably any convenient erasable block size within a chosen flash memory device, such as 8 or 16. Each sector 102 has an associated physical block address (PBA) by which it can be addressed within the mass storage memory 100 for reading, writing or erasing.

20

Each sector 102 is configured to include several fields. In the preferred embodiment, each sector includes a data field 104 for storing binary information, an error correction field 106 for correcting errors in any known manner, a logical block address LBA 108, and a plurality of flags 110. The LBA 108 is the address assigned to the corresponding data field 104 by the computer system to which the mass storage memory 100 is coupled. The flag field 110 includes at least four subfields: a used flag 120 to indicate whether a sector 102 contains information; an old flag 118 to indicate whether the data within a sector 102 has been replaced; a defect flag 122 to indicate whether the sector

25

30

102 is defective; and a system sector flag 124 to indicate whether the cluster 116 is assigned to store a system file.

In the preferred embodiment of the present invention, the old flag 118 is set for the first sector 102 within the cluster when all of the sectors 102 within the cluster have been used and the data within the last sector has been updated and stored in a first sector of a free sector. Also in the preferred embodiment, the defect flag 122 is set for the first sector 102 within the cluster if any of the sectors within the cluster are determined to be defective. When the defect flag 122 is set, the entire cluster is deemed defective and none of the sectors 102 within the cluster are used.

10 In an alternate embodiment, the old flag 118 can be set for each sector 102 as it is replaced. Also, the defect flag 122 can be set on a sector by sector basis, only for those sectors which are determined to be defective.

Each time the computer updates the system sector associated with a predetermined LBA, it selects the cluster 116 associated with that LBA. Rather than reading the old LBA data, erasing that sector 102, and then restoring the data with the updated information it merely writes the updated information into a next empty sector 102 within the cluster 116. When data is written to the next empty sector 102 within the cluster 116, the used flag 120 is set for that sector. The previous location is known to be old because the last location in the cluster 116 which contains information is the most recent location. The old flag 118 is not set until all of the sectors 102 within the cluster contain data which has been replaced. After the cluster 116 becomes full and the system tries to update the same LBA, the old flag 118 of the first sector within the cluster 116 is set so that the cluster can be subsequently erased and a new free cluster 116 is assigned to the given LBA of the system file. In this manner, an erase operation is not necessary every time a system sector is written. An erase operation of clusters having their old flags set is performed when necessary or in the background by the system.

According to the present invention, the sectors 102 of the mass storage memory 100 are partitioned into at least two groups. As shown in Figure 2, a first group 112 of the sectors 102 are reserved for exclusive use as system files. A second group 114 of the sectors 102 are reserved for use as user files and as an extension of the system area.

As is well known, system files are conventionally assigned predetermined LBAs by the computer system. For example, in DOS the first several sectors are reserved for the

operating system. In the present invention, a cluster of sectors is reserved for each LBA needed by the operating system for a system file. According to one embodiment of the present invention, if N sectors are needed, then N clusters plus an appropriate number of spares will be reserved in the first group 112.

5 Figure 3 shows a schematic representation of the clusters 116. Each one of the clusters 116 stores a sector associated with a single LBA. Each cluster 116 includes M sectors 102. Only one of the sectors 102 in each cluster 116 will contain the current information. Each sector 102 within each cluster includes the fields illustrated in Figure 1.

10 In one embodiment, such as shown in Figure 3, once the cluster is full the cluster must be marked "old" and another free cluster found before the system sector for that LBA can be updated. This procedure allows those clusters associated with the more often used system sectors to be distributed amongst the clusters assigned to the system area in order that the erase cycles are distributed more evenly among the system clusters.

15 Figure 4 shows a schematic representation of the cluster arrangement within the system partition according to the preferred embodiment of the present invention. Each cluster is assigned a PBA from PBA0 to PBA(N+...). In particular, the first group 112 (Figure 2) includes extra clusters in excess of the N required by the computer system. Each LBA 0 through N for the system sectors is assigned to a corresponding cluster PBA 0 through N. Consider for example that the system sector associated with the LBA2 is
20 written more often than the other system sectors. Once the cluster 116 for the LBA2 is filled with old system file data, it then is re-directed to an empty available cluster, eg., for PBA(N+1), as shown. The old flag for the first sector of the cluster PBA2 is set to indicate that the sectors within the cluster all contain old data and the cluster PBA(N+1) is assigned to be LBA2.

25 If the system sector associated with LBA5 then fills, the system files for that LBA are then re-directed to another empty available cluster, eg., for PBA(N+2), as shown in Figure 5. If the system sector associated with LBA2 then fills, the system files for that LBA are then re-directed to another empty available cluster, eg., for PBA(N+3), as shown in Figure 6. Once the clusters are filled, the 'old' clusters are erased and reused in this same
30 way. The system can also perform an erase of all the 'old' clusters in the background, while another operation is being performed, so it can have an empty cluster available to it at all times. This would further increase the performance of the overall system. This procedure

provides for automatically ensuring that the program and erase cycles in the system file partition of the semiconductor non-volatile mass storage memory remains approximately constant for all the sectors and clusters.

To speed up the operation of the present invention, a block RAM 134 is included
5 and used to store a table representing the LBAs, corresponding PBAs and relevant data. Figure 7 shows a schematic representation of the table stored within the block RAM for the present invention. The block RAM table includes an entry 132 for each LBA in the system file partition plus some spare locations for increased performance. The entries for each LBA 0 through N are initialized at power up to a corresponding cluster PBA 0 through N
10 by reading the LBA field 108 in the flash extension. An entry for each PBA of the spare clusters is also included within the table. Each entry 132 includes a latest physical block address of the cluster (PBA) field 134 of the information associated with an LBA as well as a mirror of the old flag 138, the used flag 140, and the defect flag 142 associated with the actual physical block address of the cluster in the semiconductor non-volatile mass storage
15 memory. Also, the field 136 represents the most recent data or sector within the given physical block address associated with the given LBA. Therefore, if the block size (cluster size) is 8, and for example the most recent sector within this cluster is 5, the field 136 is loaded with the value 5. This will help the controller during an access to locate the most recent sector within the cluster. Alternatively, this field can also be removed if the
20 controller reads every sector within the cluster to locate the most recent system file.

Within the table 144 each entry within the fields 134 and 136 corresponds to an LBA. The PBA field 134 contains the current PBA of the cluster 116 which includes the current data for the corresponding LBA. The field 136 contains the number of the sector within the cluster which is storing the current data. The flag fields 138, 140 and 142
25 correspond to the assigned PBA for each entry. The old flag field indicates that the assigned PBA for that entry is old and needs to be erased. When an erase operation is performed, the system of the present invention searches the table within the block RAM for PBAs with their old flags set. The used flag field 140 mirrors the used flag 120 of the first sector within the cluster corresponding the PBA and is set when data is stored within that
30 cluster. The defect flag field indicates that the assigned PBA cluster is defective.

Figure 8 illustrates an example of the host system accessing several LBA's and how it is handled in the table stored within the block RAM. For example, if initially the host

writes all the LBAs 0 through N, and if the mass media is not used, the LBA0 will point to PBA0, LBA1 to PBA1, . . . and LBA_N to PBA_N as shown in the table 150. For each of the entries corresponding to the LBAs 0-N, the used flag 140 within the table 150 is set. The most recent location in the cluster field 136 for each entry in the table corresponding to the LBAs 0-N will be loaded with all zero's since the 1st location in each block is the most recent system data file. Each of the tables 150-158 illustrate a progression of the entries within the table stored within the block RAM in response to a series of operations. If the host system performs a write operation to the LBA2 again, the data will be written to the next sector within the cluster PBA2 and the most recent location in the cluster field 136 within the table 151 is updated to 001. The used flag 120 will also be set in the sector extension in the non-volatile memory 100. As the system keeps writing to the LBA2, this process continues until the most recent location in cluster field 136 is updated to 111 as shown in the table 157. Then, the next time the host system performs a write operation to the LBA2 again, the old flag 118 of the first sector within the cluster PBA2 is set to 1, the old flag within the old flag field 138 corresponding to the PBA2 is set to 1, the PBA field 134 corresponding to the entry LBA2 in the table 158 will be updated with the next available free cluster, which is PBA(N+1) and the used flag field 140 corresponding to the entry for the free cluster PBA(N+1) is set to 1. Then, the most recent location in the cluster field 136 is reset to 000. The next available free cluster is found by searching the clusters for a cluster which includes a first sector in which the used and defect flags are not set. When a cluster is erased the old and used flags are reset within the cluster 116 and within the table 144. This cluster is then added to the pool of free clusters.

A control circuit, as illustrated in Figure 9, is configured for reading the flags and receiving data requests from the computer system for controlling the operations of the semiconductor non-volatile mass storage memory described herein. The control circuit 200 is coupled to the mass storage media 212, to the host system 210 and to a local controller 214 for controlling the operations between the host system 210 and the mass storage media 212. The control circuit 200 includes a media control circuit 202, a controller state machine 204, one or more host state machines 206 and a block RAM 208. The media control circuit 202 controls and provides the interface to the mass storage media 212. The host state machine 206 controls and provides the interface to the host system 210. The controller state machine 204 controls the operation of the control circuit 200. The block

RAM 208 is used to store the table 144, as described above.

The table 144 is stored within the block RAM 208 which is volatile memory. Therefore, when the power to the system is turned off, the data stored within the table 144 is lost. Accordingly, when power is restored to the system the data has to be restored or
5 reloaded into the table 144. In order to reload the data within the table 144, each cluster which includes a system sector is read and the data from each sector is used to build the table 144. Only system clusters in which the old flag 118 and/or the defect flag 122 of the first sector is not set, are read. Each sector includes an LBA field 108 which is used to correlate a PBA with each LBA. This PBA is then stored within the PBA field 134 of the
10 table 144 for a corresponding LBA entry. The entry for the most recent location in the cluster field 136 is determined by finding the last sector in the cluster which has its used flag 120 set. In this manner, this process is repeated for each current cluster storing a system sector until the table 144 is rebuilt.

The present invention has been described in terms of specific embodiments
15 incorporating details to facilitate the understanding of the principles of construction and operation of the invention. Such reference herein to specific embodiments and details thereof is not intended to limit the scope of the claims appended hereto. It will be apparent to those skilled in the art that modifications may be made in the embodiment chosen for illustration without departing from the spirit and scope of the invention.

CLAIMS

1. A semiconductor non-volatile mass storage memory, wherein the semiconductor non-volatile mass storage memory comprises a plurality of individually addressable sectors for storing data, the semiconductor non-volatile mass storage memory comprising:
 - a. a first partition in the semiconductor non-volatile mass storage memory having a plurality of sectors for storing user files; and
 - b. a second partition in the semiconductor non-volatile mass storage memory for storing system files, wherein the second partition further comprises a plurality of clusters, each cluster having a plurality of sectors and each cluster for storing a system sector for a predetermined logical block address within one of the sectors in that cluster; and
 - c. a control circuit for writing a replacement system sector into an empty sector within the cluster and for providing an indicator to a replaced system sector.
2. The semiconductor non-volatile mass storage memory according to claim 1 further comprising an empty cluster for receiving a subsequent replacement system file when a previous cluster is filled.
3. The semiconductor non-volatile mass storage memory according to claim 2 wherein each of the entries also contains flags indicative of whether a sector associated with the logical block address is old.
4. The semiconductor non-volatile mass storage memory according to claim 3 wherein each of the sectors also contain a flag indicative of whether the sector is used.
5. The semiconductor non-volatile mass storage memory according to claim 3 wherein each of the sectors also contain a flag indicative of whether the sector is defective.
6. A memory system for storing data comprising:
 - a. a semiconductor non-volatile mass storage memory including a partition for storing system files having a plurality of clusters, each cluster including a plurality of

- sectors and each cluster for storing a system sector for a single logical block address within one of the sectors in that cluster; and
- b. a control circuit coupled to the semi-conductor non-volatile mass storage memory for controlling memory access operations to and from the semiconductor non-volatile mass storage memory.
7. The memory system as claimed in claim 6 wherein a write operation is completed by writing a replacement system sector into an empty sector within the cluster corresponding to an appropriate logical block address.
8. The memory system as claimed in claim 7 wherein the semiconductor non-volatile mass storage memory further comprises an empty cluster for storing a subsequent replacement system file when a previous cluster is filled.
9. The memory system as claimed in claim 8 further comprising a table coupled to the control circuit and to the semiconductor non-volatile mass storage memory for storing information relating to a current location of each logical block address, wherein the table includes a plurality of addressable entries, each of the entries corresponding to a logical block address, and further wherein each of the entries is configured to contain a corresponding physical block address of a sector.
10. The memory system as claimed in claim 9 wherein the table is stored in a random access memory.
11. The memory system as claimed in claim 10 wherein the table is restored by reading each of the clusters corresponding to a logical block address.
12. The memory system as claimed in claim 11 wherein each of the entries within the table contains a flag indicative of whether a cluster corresponding to the logical block address is used.

13. The memory system as claimed in claim 12 wherein each of the entries within the table contains a flag indicative of whether a cluster corresponding to the logical block address is defective.
14. The memory system as claimed in claim 11 wherein each of the entries within the table further includes an indication of a current sector within the cluster corresponding to the logical block address.
15. A method of storing information in a semiconductor non-volatile mass storage memory comprising:
- a. partitioning the memory into a first group having a plurality of sectors for storing a plurality of user files and a second group having a plurality of sectors for storing a plurality of system files;
 - b. partitioning the second group into clusters, each cluster having a plurality of sectors and each cluster for storing a system sector for a single logical block address within one of the sectors in that cluster;
 - c. determining which sector in that cluster is an empty sector;
 - d. writing a replacement system file into the empty sector;
 - e. providing indicators to the current and replaced system sectors; and
 - f. marking a cluster as old when a last sector within the cluster is replaced and there is no free sector within the cluster.
16. The method according to claim 15 further comprising the step of writing a subsequent replacement system sector into an empty cluster when a previous cluster is filled.
17. The method according to claim 16 further comprising the step of erasing any clusters having only old system sectors.

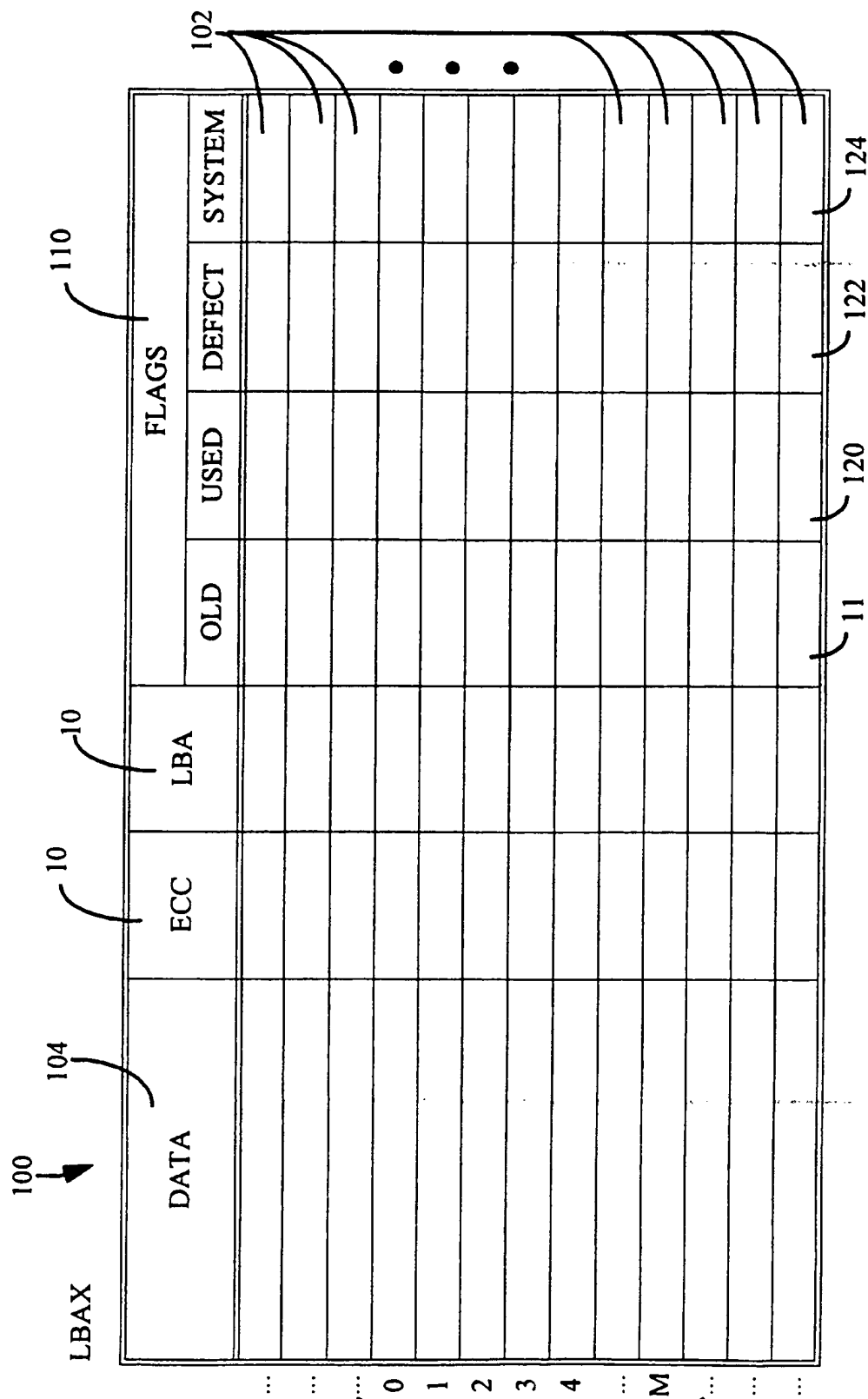
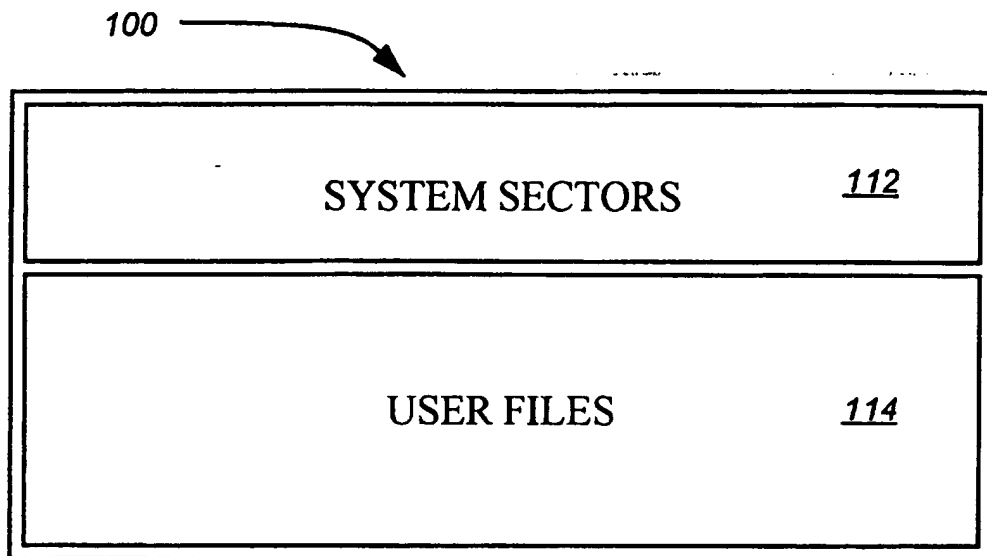
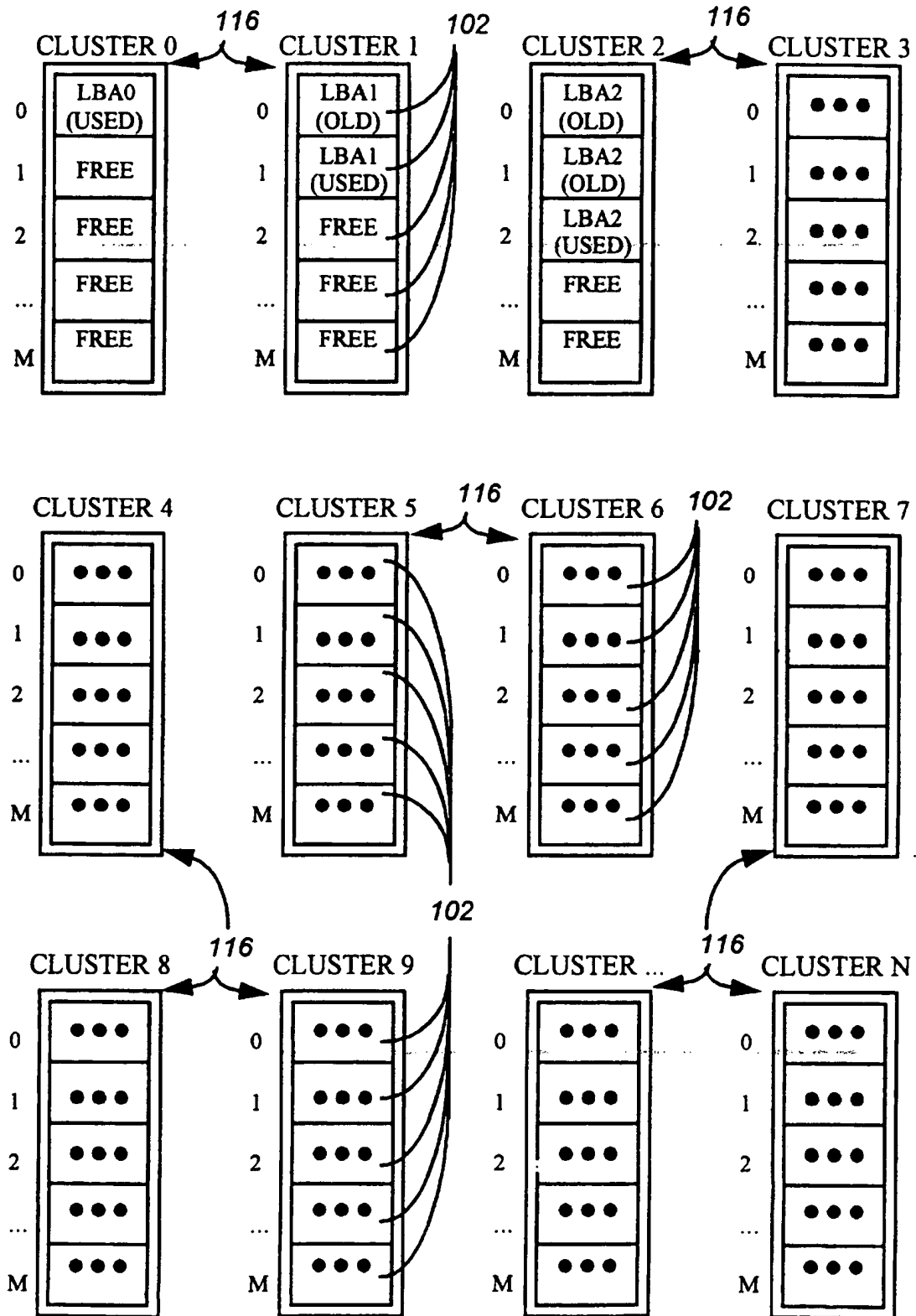
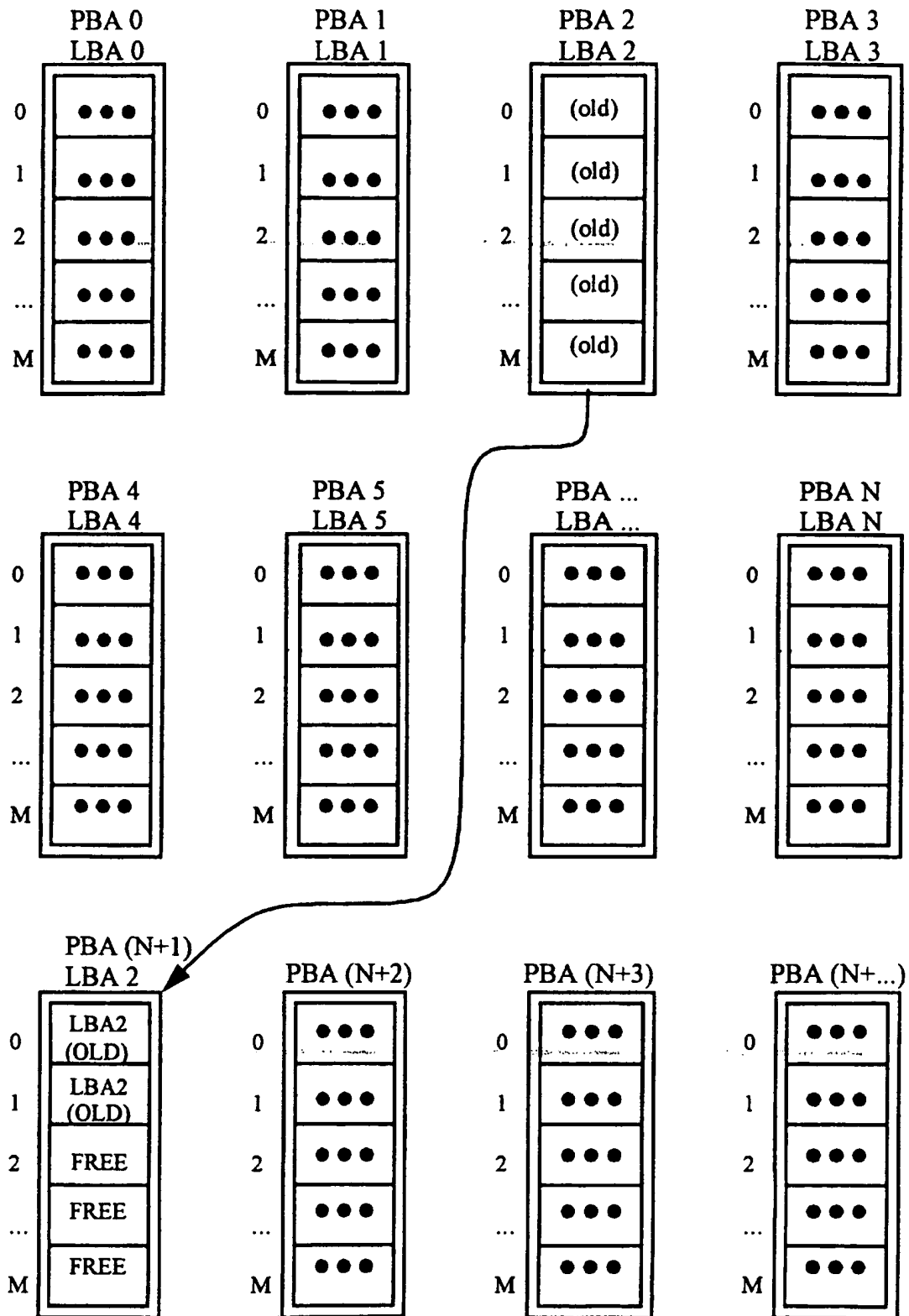
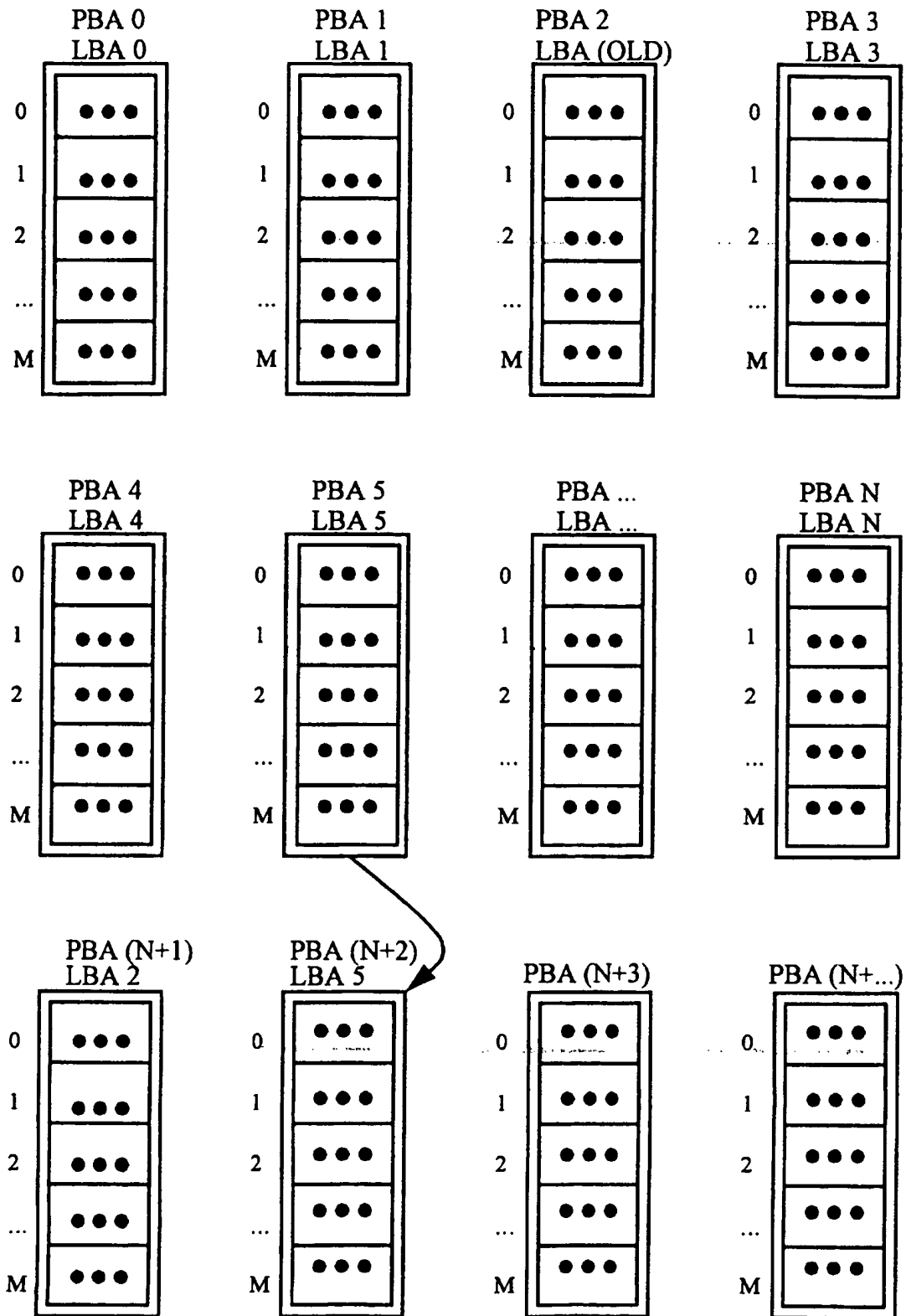


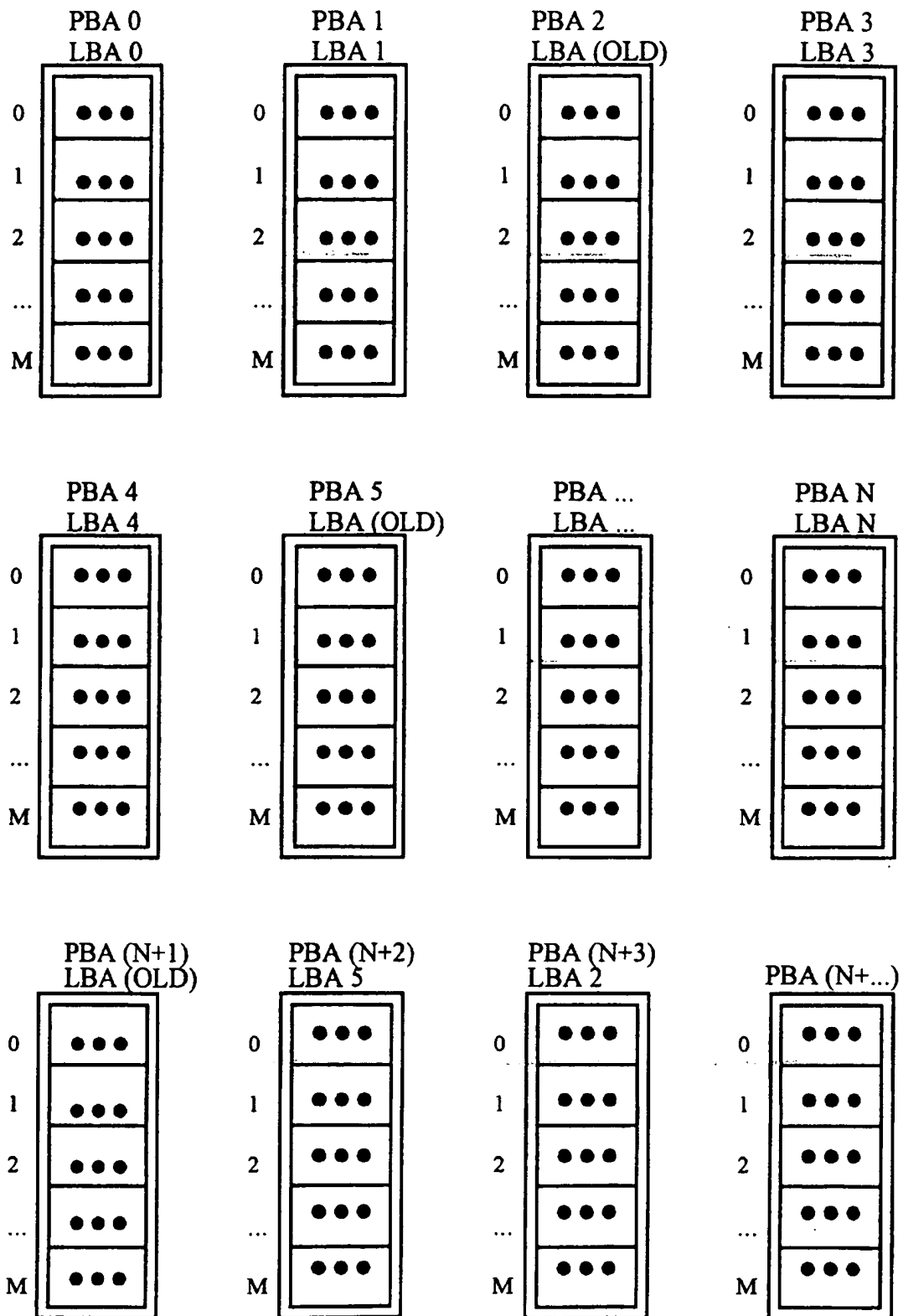
FIGURE 1

**FIGURE 2**

**FIGURE 3**

**FIGURE 4**

**FIGURE 5**

**FIGURE 6**

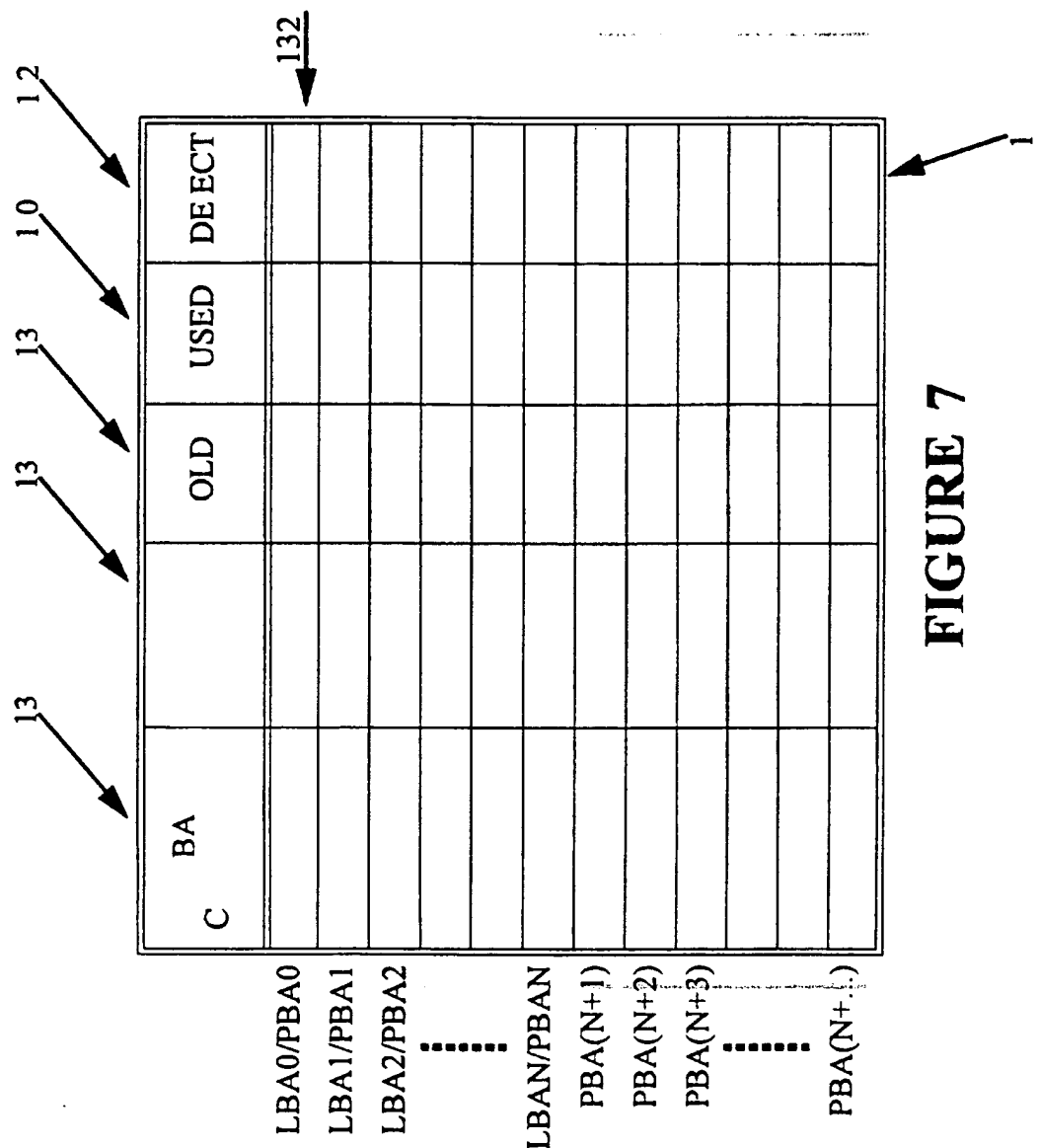


FIGURE 7

150

134 136 138 140 142

	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	000	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

151

	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	001	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

152

	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	010	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

FIGURE 8A
8/11

153 → 134 → 136 → 138 → 140 → 142

	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	011	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

154 →


	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	100	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

155 →


	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	101	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

FIGURE 8B

	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	110	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

157 

	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	111	0	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	0	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

158 

	PBA Cluster location	most recent location in cluster	OLD	USED	DEFECT
LBA0/PBA0	PBA0	000	0	1	0
LBA1/PBA1	PBA1	000	0	1	0
LBA2/PBA2	PBA2	000	1	1	0
...					
LBA5/PBA5	PBA5	000	0	1	0
...					
LBAN/PBAN	PBAN	000	0	1	0
PBA(N+1)	X	000	0	1	0
PBA(N+2)	X	000	0	0	0
PBA(N+3)	X	000	0	0	0
	X	000	0	0	0
PBA(N+...)	X	000	0	0	0

FIGURE 8C
10/11

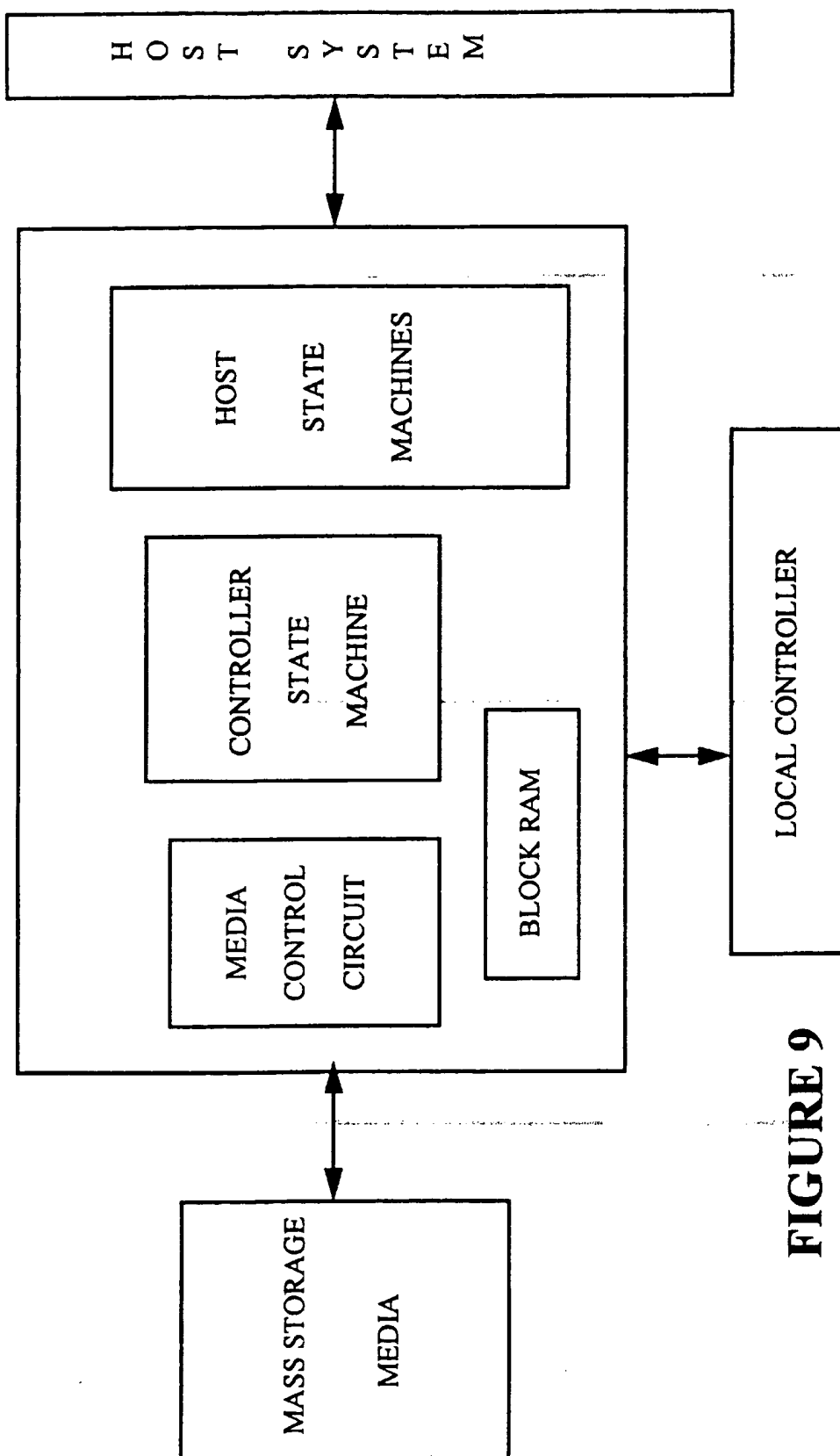


FIGURE 9

INTERNATIONAL SEARCH REPORT

Int. Application No
PCT/US 96/14750

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G11C16/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G11C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	EP,A,0 686 976 (FUJITSU LTD) 13 December 1995 see abstract see column 6, line 7 - line 27 see column 6, line 47 - line 50 see column 8, line 15 - line 36; figures 6,7	1-17
A	JP,A,06 250 798 (IBM) 9 September 1994 & US,A,5 457 658 (NIJIMA HIDETO ET AL) 10 October 1995 see abstract see column 3, line 30 - line 46 see column 5, line 35 - column 7, line 54 --- -/--	1-17

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the international search

10 December 1996

Date of mailing of the international search report

19.12.96

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+ 31-70) 340-3016

Authorized officer

Gonzalez Ordóñez, O.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 96/14750

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,A	<p>US,A,5 544 356 (ROBINSON KURT B ET AL) 6 August 1996 see column 4, line 19 - line 46 see column 7, line 19 - line 52 see column 16, line 1 - line 26 -----</p>	1-17

INTERNATIONAL SEARCH REPORT

Information on patent family members

In ternational Application No

PCT/US 96/14750

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0686976	13-12-95	JP-A- 5151097	18-06-93
		EP-A- 0544252	02-06-93
		US-A- 5530827	25-06-96

JP-A-6250798	09-09-94	US-A- 5457658	10-10-95

US-A-5544356	06-08-96	FR-A- 2671223	03-07-92
		GB-A,B 2251324	01-07-92
		JP-A- 6139140	20-05-94
